H19/B09 証明論的アプローチによるプログラム構成原理(1節 共同プロジェクト研究の理念と概要,第4章 共同プロジェクト研究)

雑誌名	東北大学電気通信研究所研究活動報告
巻	15
ページ	226-228
発行年	2009-08
URL	http://hdl.handle.net/10097/48428

共同プロジェクト研究

課題番号　Ｈ１９／Ｂ０９

採択回数	１　②　３

証明論的アプローチによるプログラム構成原理

［１］組織
代表者：佐藤 雅彦
（京都大学大学院情報学研究科）
対応者：外山 芳人
（東北大学電気通信研究所）
分担者：
五十嵐 淳
（京都大学大学院情報学研究科）
中澤 巧爾
（京都大学大学院情報学研究科）
亀山 幸義
（筑波大学システム情報工学研究科）
桜井 貴文（千葉大学理学部）

研究費：物件費 17,899 円，旅費 238,760 円

［２］研究経過

本プロジェクトの目的

証明の構文的性質を議論する証明論は，論理学における主要な分野の一つとして古くから活発に研究されているが，構文的対象を操作するその手法は，計算機科学，とくにプログラムとその計算過程を構成的な立場から構文論的に表現する操作的意味論と密接に関連する．本研究では，証明論的手法をプログラム意味論，とくに操作的意味論の研究に応用し，構成的な立場からより性質のよいプログラミング言語を設計する指針を与えることを目的とする．

研究会の開催状況

本プロジェクトは本年度が第二年度であった．上記目的達成のため，一回のプロジェクト研究会を開催し，活発な議論を通して，ロジックの立場，計算論の立場，さらにこれらを統合した型理論の立場からプログラミング言語の本質を追及した．

プロジェクト研究会は，平成 20 年 11 月 19 日から 20 日に東北大学電気通信研究所において開催され，プロジェクトの成果報告が行なわれた．

［３］成果

（３－１）研究成果

本年度は，以下に示す研究成果を得た．

1. 数学を形式的に記述するときに基本的な抽象(abstraction)操作について考察した．この操作は，与えられた変数 x と表現 M から表現 abs(x, M) を構成する操作であり，その結果は M の x による抽象とよばれる．この操作の「逆操作」は具体化(instantiation)とよばれ，抽象 A を表現 N で具体化した結果は inst(A, N) とかかれる．抽象と具体化の間には以下の関係がある：inst(abs(x, M), N) = [N/x]M．これらの性質をもつ抽象操作を実現する具体的な新しい方法を提案した．

2. 等式論理の帰納的定理として形式化したプログラムの性質の自動検証においては，適切な補題の発見が証明成功の１つの鍵である．このため，さまざまな補題の自動生成法が提案されている．帰納的定理の自動証明法における補題の自動生成法の１つで

ある健全一般化法について，従来与えられていた条件の誤りを発見し反例を与えるとともに，その他の条件の一部を緩和し，より広範囲に適用可能な健全一般化法を提案した.

3. 合流性は計算モデルの基本的な性質として重要な性質である．項書き換えシステムの合流性検証システムとして，分割統治型の合流性検証システムを設計・実現した．我々の合流性検証システムは，合流性を保証する複数の条件の検証と，部分システムの合流・非合流性から全体の合流・非合流性を判定する合流性のモジュラー性に基づく．これにより，単一の条件では合流性の判定が困難な例に対しても，複数の条件を組み合わせた合流性の検証が可能である.

4. 古典論理に基づく型体系から直観主義論理に基づく型体系への簡約を厳密に保存する(つまり 1 つの簡約が 1 つ以上の簡約に写す)変換を与えた。扱った古典論理型体系は，lambda-bar-mu，lambda-mu，call-by-name lambda-bar-mu-mu-tilde，call-by-value lambda-bar-mu-mu-tilde である．このような変換は他にもあるが，本研究では call-by-name lambda-bar-mu-mu-tilde の簡約を保存する変換は初めて与えられたものであり，いくつかの知られている変換は簡単な変換に分解して得られることを示した．また，これら変換により古典論理型体系の強正規化性を直ちに導くことができる.

5.「プログラムを生成するプログラム」を記述する言語であるマルチステージプログラミング(MSP)言語は，生成されたプログラムの型安全性が保証されるという特徴を有する一方，純粋なラムダ計算の範囲に限定されている，という問題点があった．今年度の研究では，実行効率の良いプログラムの生成で必要となる，制御演算子を導入した MSP 言語を設計し，型安全性や主型の存在など基礎理論を構築した．これにより，計算の効果(エフェクト)を利用した MSP が可能であることを明らかにした。

6. メタプログラミング言語の基盤的計算体系として提案されている Taha と Nielsen の λ^{α} と，その中に現れる環境分類子の概念に対する，カリーハワード同型対応を用いた論理的意味付けを行った．その結果，λ^{α} を修正することにより， 時間遷移列に関する量化子を持つ多次元線形時間時相論理とでも言うべき論理に対応し，環境分類子は時間遷移列上を動く変数として捉えられることがわかった.

7. 型付きラムダ計算に対する縮約可能性法を用いない強正規化性の証明を，合流性や標準化定理等，他の簡約の性質にも適用することに取り組んだ．これらの証明はシーケント計算の体系に基づく型システムを用いて与えることができる．また，それらの方法をインターセクション型システムに対する体系にも拡張し，型付きラムダ項の性質を証明するための手法として，本研究の方法が一般的かつ統一的な手法であることを示した.

8. 値呼び計算と名前呼び計算の双対性について分析をし，従来は等号関係に基づく計算体系で知られていた双対性を簡約関係に基づく計算体系に強められることを証明した．これは Wadler が与えた双対計算とラムダミュー計算の等号体系を用いて証明した手法を，計算に非本質的な部分を除去し、簡約体系に置き換えることで行われた. これにより Wadler が 2005 年に未解決として提案した問題に答えた.

9. 値呼びラムダ計算の CPS 変換として，その操作的意味に関して完全であるものを与えた. 既に等号理論によって与えられる意味において完全な CPS 変換は知られていたが，本研究では，新たに CPS 言語を定義することにより，操作的意味，すなわち簡約関係について完全であるような CPS 変換を定義した．また，この議論が値呼びラムダミュー計算に対して自然に拡張できることを示した.

（3－2）波及効果と発展性など

本プロジェクト研究では，より性質の良いプログラミング言語の実現を目標に，主に構成的な立場からプログラム意味論の基礎的な研究を行ない，関連の成果が蓄積されつつある．構文的対象を直接扱う構成的な手法に基くアプローチは，計算機上での実現が比較的容易であると考えられる．このため本研究の成果は純粋な理論的成果に留まらず，プログラム検証や開発支援技術の新しい可能性を開くものとして期待できる．また，本プロジェクトの活発な交流を通して形成された，複数の大学の多岐に渡る計算機科学分野の研究グループ間のネットワークは，今後の電気通信研究所を中心とした国際的な研究ネットワークを形成する基盤となることが期待される.

［４］成果資料

[1] Masahiko Sato. A framework for checking proofs naturally. Journal of Intelligent Information Systems, vol. 31, 111—125, 2008.

[2] Masahiko Sato. External and internal syntax of the lambda-calculus. In Buchberger, Ida, Kutsia (Eds.), Proc. of the Austrian-Japanese Workshop on Symbolic Computation in Software Science, SCSS 2008, 176 — 195, 2008.

[3] Takahito Aoto. Sound lemma generation for proving inductive validity of equations. In Proceedings of the 28th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2008), Bangalore, India, December 2008, pp. 13-24, Dagstuhl Seminar Proceedings, Vol. 08004, IBFI, Schloss Dagstuhl, Germany.

[4] Takahito Aoto. Designing a rewriting induction prover with an increased capability of non-orientable theorems. In Proceedings of Austrian-Japanese Workshop on Symbolic Computation in Software Science (SCSS 2008), Hagenberg, Austria, July 2008, pp. 1-15, RISC Technical Report 08-08.

[5] Yukiyoshi Kameyama, Takuo Yonezawa. Typed Dynamic Control Operators for Delimited Continuations. Proc. International Symposium on Functional and Logic Programming (FLOPS 2008), Lecture Notes in Computer Science 4989, pp. 239-254, Apr. 2008.

[6] Yukiyoshi Kameyama and Kenichi Asai. Strong Normalizatoin of Polymorphic Calculus for Delimited Continuations. Proc. Symbolic Computation in Software Science (SCSS 2008), RISC-Linz Report Series No. 08-08, pp. 96-108, Hagenberg, Austria, July. 2008.

[7] Jefferson O. Andrade and Yukiyoshi Kameyama. A Direct Algorithm for Multi-Valued Bounded Model Checking. Proc. International Symposium on Automated Technology for Verification and Analysis (ATVA 2008), Lecture Notes in Computer Science 5311, pp. 80-94, Oct. 2008.

[8] Yukiyoshi Kameyama, Oleg Kiselyov, and Chung-chieh Shan. Shifting the Stage — Staging with Delimited Control. Proc. ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation (PEPM'09), Savannah, USA, pp. 111-120, Jan. 2009.

[9] Naokata Shikuma and Atsushi Igarashi. Proving noninterference by a fully complete translation to the simply typed λ-calculus. Logical Methods in Computer Science, 4(3:10):1-31, September 2008.

[10] Kensuke Kojima and Atsushi Igarashi. On constructive linear-time temporal logic. In Proceedings of the Intutionistic Modal Logics and Applications Workshop (IMLA'08), Pittsburgh, PA, June 2008.

[11] Takeshi Tsukada and Atsushi Igarashi. A Logical Foundation for Environment Classifiers. To appear in Proceedings of 9th International Conference on Typed Lambda-Calculi and Applications (TLCA 2009).

[12] Kentaro Kikuchi. Call-by-Name Reduction and Cut-Elimination in Classical Logic, Annals of Pure and Applied Logic, Vol. 153, No. 1-3, pp. 38-65, 2008.

[13] Kentaro Kikuchi and Stéphane Lengrand. Strong Normalisation of Cut-Elimination that Simulates β-Reduction, in Proceedings of the 11th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2008), LNCS 4962, pp. 380-394, 2008.

[14] 木村大輔, 角谷良彦. 古典様相論理に対応する計算体系. 第11回プログラミングおよびプログラミング言語ワークショップ論文集, pp. 23-37, 2009年3月.

[15] Koji Nakazawa, Makoto Tatsuta, Yukiyoshi Kameyama, and Hiroshi Nakano. Undecidability of type-checking in domain-free typed lambda-calculi with existence. In Michael Kaminski and Simone Martini, editors, Computer Science Logic (CSL 2008), volume 5213 of Lecture Notes in Computer Science, pages 478 - 492. Springer-Verlag, 2008.

[16] Koji Nakazawa and Makoto Tatsuta. Type checking and inference for polymorphic and existential types. In Computing: The Australasian Theory Symposium (CATS 2009), January 2009.